Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Engineering 42 (2012) 106 – 112
Procedia Engineering
www.elsevier.com/locate/procedia

20th International Congress of Chemical and Process Engineering CHISA 2012
25 – 29 August 2012, Prague, Czech Republic

An optimized neural network model of desalination by vacuum membrane distillation using genetic algorithm

M. Tavakolmoghadam[a] a*, Mohammadali Safavi[a,b]

[a]*Research Institute of Petroleum industry, P.O. Box 14665-137, Tehran, Iran*
[b]*Chemical & Petroleum Engineering Department, Sharif University of Technology, Azadi Ave., Tehran, Iran*

Abstract

An experimental based ANN model is constructed to describe the performance of vacuum membrane distillation process for desalination in different operating conditions. The vacuum pressure, the feed inlet temperature, the concentration of the feed salt aqueous solution and the feed flow rate are the input variables of this process, whereas the response is the permeate flux. The neural network approach was found to be capable for modeling this membrane distillation configuration. The application of Genetic Algorithm (GA) to optimize the ANN model parameters was also investigated.

Keywords: Vacuum membrane distillation, Artificial neural network, Desalination, Optimization, Genetic Algorithm

1. Introduction

Mathematical models for prediction of membrane separation play an important role in optimization of membrane systems leading to efficient and economical design of separation processes. Artificial neural networks (ANN) modeling was applied in different areas of membrane science and technology such as in

* Corresponding author. Tel.: *+98-2-148253109*.
E-mail address: *tavakolm@ripi.ir*

doi:10.1016/j.proeng.2012.07.400

the pressure-driven membrane processes, microfiltration, ultrafiltration, nanofiltration, and reverse osmosis, due to its potential to study the relationship between the input variables and the target(s) or output(s) of the process using a limited number of experimental runs [1].

The model developed by ANN could be considered as the fitness function for optimization by genetic algorithm (GA). Recently, ANN (combined with GA) has become a popular approach to solve optimization problems in many processes without theoretical or mechanistic dependence [2-4].

Among various membrane desalination processes, RO or membrane distillation (MD) is believed to have a great potential for the production of drinking water from seawater and brackish water [5]. MD differs from other membrane technologies in that the driving force for desalination is the difference in vapor pressure of water across the membrane, rather than total pressure. A variety of methods have been employed to impose the vapor pressure difference across the hydrophobic membranes. In every case, the water to be desalted directly contacts the hot side of the membrane. Generally, there are four different techniques and configurations of the MD processes: direct contact membrane distillation (DCMD), air gap membrane distillation (AGMD), sweeping gas membrane distillation (SGMD) and vacuum membrane distillation (VMD) [6].

In this study an experimental based ANN model is constructed to describe the performance of vacuum membrane distillation process for desalination in different operating conditions. The application of Genetic Algorithm (GA) to optimize the ANN model parameters was also investigated.

2. Experimental

The VMD experiments were carried out using the experimental set-up and procedure presented in references No. 6 and 7. Experiments were carried out using a flat sheet Polytetrafluoroethylene (PTFE) membrane from Membrana (Germany). A cross flow membrane module made from Teflon was used in the experiments. Membrane properties are reported in Table 1.

Table 1. Properties of the flat sheet PTFE membrane

Parameter	Amount
Pore size, μm	0.2
Porosity, %	80
Thickness, μm	60

3. Modeling and optimization

3.1. ANN Modeling

ANN is a non-linear processing system operating in parallel being composed of neurons and connections between them that can be used for mapping input and output data [8]. An artificial neuron is a single computational processor, which has two operators (1) summing junction and (2) transfer function [9]. The connections consist of weights and biases with neurons addressing information. Considering the model of a single neuron, any scalar input x_i is transmitted via a connection that multiplies its strength by the scalar weight w_i to form the product w_i x x_i. The bias b is much like a weight, except that it has a constant input of unity and it is simply added to the product w_i x x_i by summing junction.

The summing junction operator of a single neuron summarizes the weights and bias into a net input λ known as argument to be processed:

$$\lambda = \sum_{i=1}^{n} x_i \,.\, w_i + b \tag{1}$$

where w_i (i = 1, n) are the connection weights, x_i is the input variable, n is the number of input variables, i is the integer index and b is called bias.

In our experiment, the ANN architecture consists of four neurons (The vacuum pressure, the feed inlet temperature, the concentration of the feed salt aqueous solution and the feed flow rate) in the input layer, three neurons in the hidden layer, and one neuron (Permeate flux) in the output layer (Fig. 1).

In this study 252 different experiments were performed applying different VMD operating conditions to develop the ANN model. The data were fed to train an ANN model using back-propagation algorithm. 66% of the total experimental data set (167 cases) was used for training that was done according to the Marquardt algorithm and remaining for validation and testing of the ANN model. The split of data into training, validation and test subsets was carried out to estimate the performance of the neural network for prediction of ''unseen'' data that were not used for training. In this way the generalization capability of the ANN model can be evaluated. The ranges of input variable are shown in Table 2. The tangent sigmoid and pure linear functions were used as the transfer functions in the hidden and output layers of the ANN, respectively. The mean square error between the results of the output neurons and the actual outputs is calculated and propagated backward through the network. Then the algorithm adjusts the weight of each. Once the mean square error got to 1e-4, the training was over and the corresponding ANN was built.

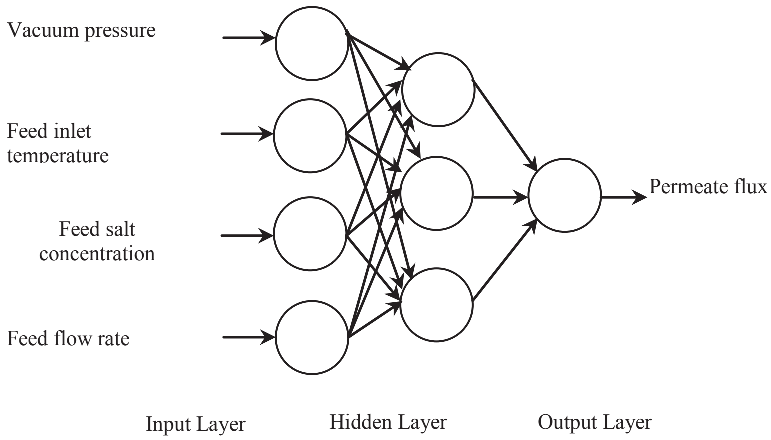

Fig.1. Schematic representation of ANN modelling

Table 2.Input variables range

Input variable	vacuum pressure (mbar)	The feed inlet temperature (°C)	The Concentration of Feed salt aqueous solution (g/l)	The feed flow rate (ml/s)
Level	10,30,50,80	25,35,45,55	50,100,200,300	15,30,45,60

Figs. 2-4 show the agreement between the experimental data and the ANN predicted results for training, validation and test data subsets with 3 neurons in the hidden layer and 75 epochs before optimization.

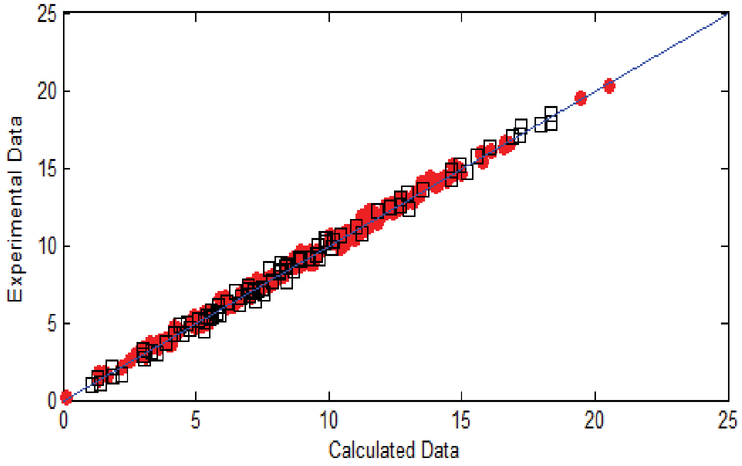

Fig. 2. The agreement between the experimental data and the ANN predicted results before optimization

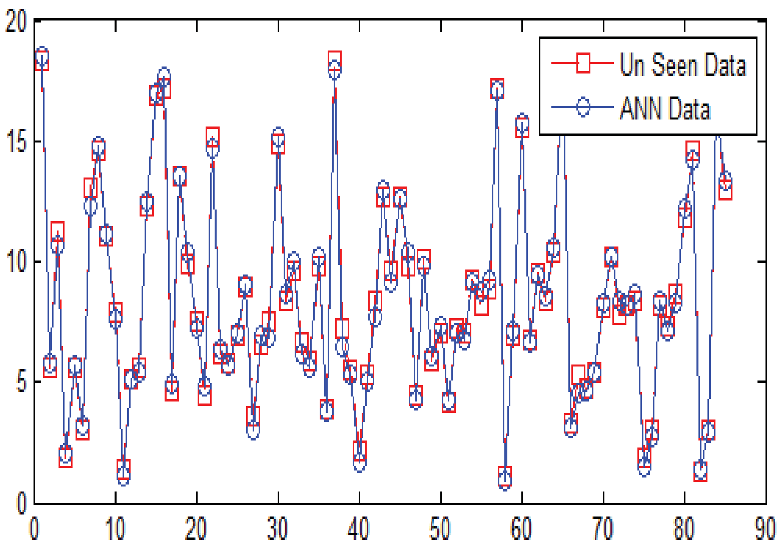

Fig. 3. The agreement between the Un-seen experimental data and the ANN predicted results before optimization

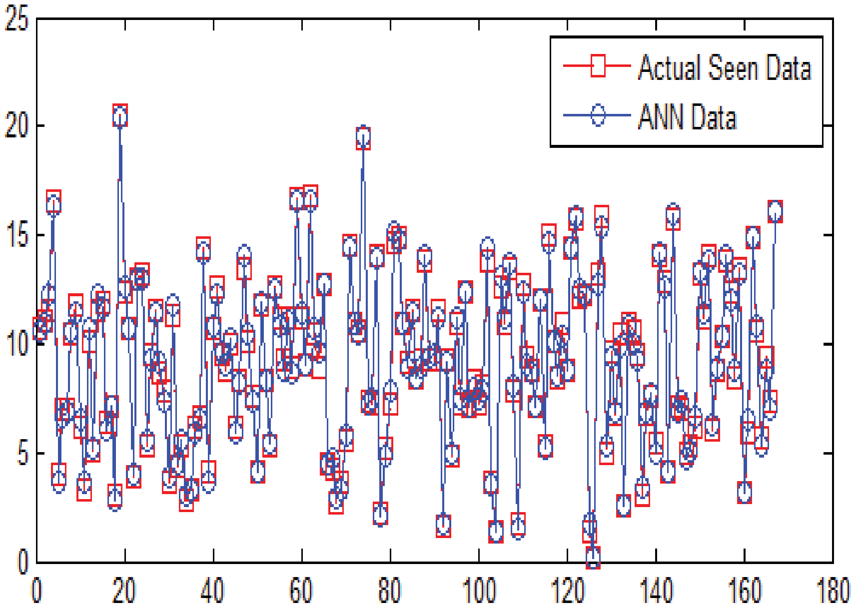

Fig. 4. The agreement between the actual-seen experimental data and the ANN predicted results before optimization.

3.2. Optimization of ANN model using GA

There are several methods that are used to select an appropriate structure of neural network for a special case. Some of these methods are based on optimization algorithms such as genetic algorithm and some of them are based on trial and error algorithms.

Genetic algorithm (GA) performs random searches through a given set for finding the best criteria of goodness. These criteria are expressed as an objective or fitness function. Fitness is defined as to be either maximized or minimized. The GAs searching ability has been enhanced by operators such as crossover, selection operator and three mutations consisting of inverse- anticodon, maximum–minimum operator, and normal-mutation [10].

In the GA, each solution to a given problem is encoded as a chromosome, which evolves over time towards a better solution. Some of the advantages of GA over the conventional optimization methods are short calculation time, flexibility, robustness and high convergence property. The use of GA requires the choice of a set of operational parameters such as population size, mutation rate and crossover rate. Crossover combines the features of two parent chromosomes (solutions) to form two new similar children (new solutions) by swapping corresponding segments of parents. Mutation rules apply random changes to individual parents to form children. Crossover is aimed at exchanging information between different potential solutions, while mutation is aimed at introducing some extra variability into the population.

In this study, using the trained ANN as the fitness function, GA was implemented to optimize the conditions for maximum. Randomly generate a population of individuals and assign a fitness value to each individual by specific fitness function. Select individuals with higher fitness values and make them undergo genetic operation such as crossover and mutation. Use the newly generated child population as the parent population for the next generation and treat them with the same evolutional process continuously until a stop criterion has been satisfied.

In order to optimize ANN model variables such as number of neurons, the number of epochs and the coefficients of the model, the Genetic Algorithm is used by linking ANN model with GA. The GA was coded separately and proposed to find the best level of training for each ANN structure. The GA was set to the population size of 80, crossover fraction of 0.9 (population fraction at the next generation that is created by the crossover function), migration fraction of 0.1 (specifying the fraction of individuals in each subpopulation that migrates to a different subpopulation), and generation of 100 and acquired stop searching by exceeding the maximum number of generations. Fig. 5-7 illustrate the fitness of the model and the predicted values.

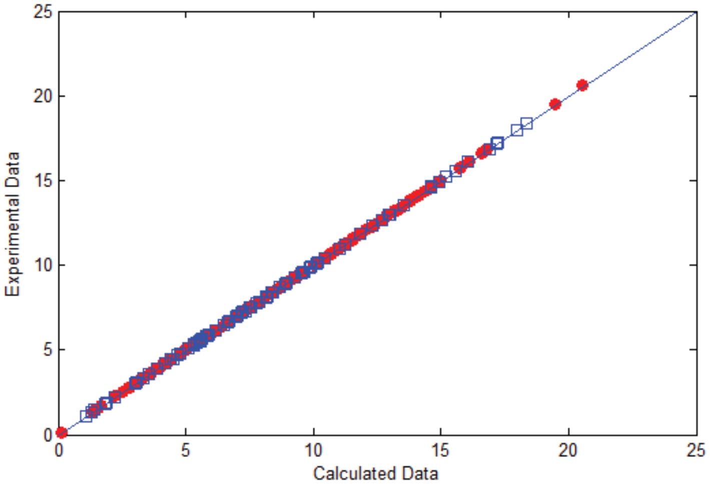

Fig. 5. The agreement between the experimental data and the ANN predicted results after optimization

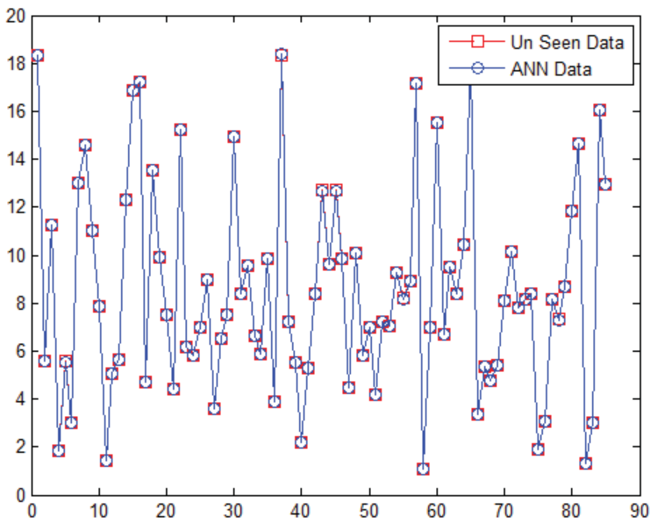

Fig. 6. The agreement between the Un-seen experimental data and the ANN predicted results after optimization

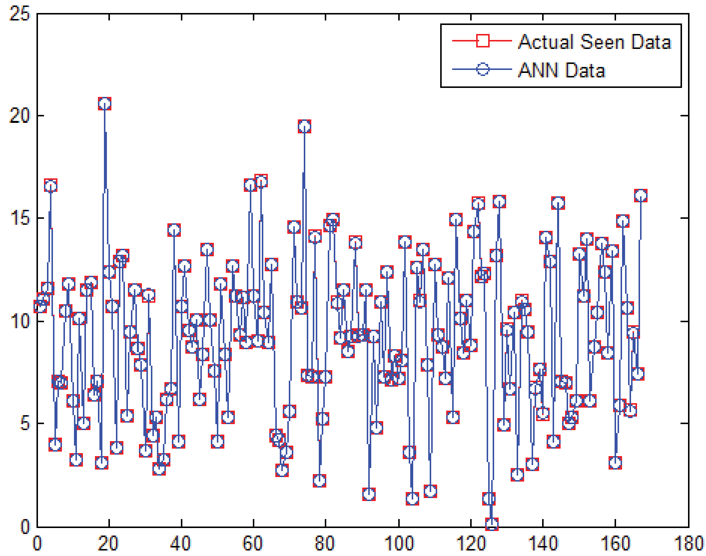

Fig. 7 The agreement between the actual-seen experimental data and the ANN predicted results after optimization

4. Results

As mentioned, a feed forward neural network with one hidden layers and back propagation training function was developed. Four variables namely vacuum pressure, the feed inlet temperature, the concentration of feed salt aqueous solution and the feed flow rate, as input parameters introduced to the ANN model.

It was found that the network composing three neurons in the hidden layer has the least number of neurons with an acceptable error (about 4%), while the network optimized according to the GA optimization, composing five neurons in the hidden layer has the least errors (less than 1%).

References

[1] Khayet M, Cojocaru C. Artificial neural network modeling and optimization of desalination by air gap membrane distillation. *Sep purif technol* 2012;**86**:171-182.

[2] Serra JM, Corma A, Chica A, Argente E,V. Botti. Can artificial neural networks help the experimentation in catalysis?. *Catal. Today* 2003;**81**:393–403.

[3] Wang, JL, Wan, W.Application of desirability function based on neural network for optimizing biohydrogen production process. *Int. J. Hyd. Energy* 2009;**34**:1253–1259.

[4] Alklaibi A M, Lior N. Membrane-distillation desalination: status and potential, Desalination 2004; 171: 111–131.

[5] Zhang Y. et al. Artificial neural network-genetic algorithm based optimization for the immobilization of cellulase on the smart polymer Eudragit L-100.Bioresource Technology 2010; 101 3153–3158.

[6] Safavi M., Mohammadi T. High-salinity water desalination using VMD, *Chem Eng J* 2009;**149**:191–195.

[7] Mohammadi T, Safavi M.A. Application of Taguchi method in optimization of desalination by vacuum membrane distillation. Desalination 2009;**249**:83–89.

[8] Sarkar B., Sengupta A., De S., DasGupta S. Prediction of permeate flux during electric field enhanced cross-flow ultrafiltration - A neural network approach. *Sep purif technol* 2009;**65**:260–268.

[9] Demuth H, Beale M., *Neural Network Toolbox: For Use with MATLAB Version 4.0*, Massachusetts: The MathWorks Inc.; 2004.

[10] Rajasekaran S,. Pai GAV. *Neural Networks, Fuzzy Logic, and Genetic Algorithms*. 7th ed. India: Prentice-Hall; 2007.